Lab	Type	Practical
Basic Python Programs		
1	A	1. WAP to print "Hello World.!"
	A	2. WAP to print addition of two numbers with and without using input().
	A	3. WAP to check the type of the variable.
	A	4. WAP to calculate simple interest.
	A	5. WAP to calculate area and perimeter of a circle.
	A	6. WAP to calculate area of a triangle.
	A	7. WAP to compute Quotient and remainder.
	A	8. WAP to convert degree into Fahrenheit and vice versa.
	A	9. WAP to find the distance between two points in 2-D space.
	B	10. WAP to print sum of n natural numbers.
	B	11. WAP to print sum of square of n natural numbers.
	B	12. WAP to concate the First and Last name of the student.
	B	13. WAP to swap two numbers.
	C	14. WAP to get the distance from user into kilometer, and convert it into meter, feet, inches and centimeter.
	C	15. WAP to get day, month and year from the user and print the date in the given format: 20-10-2024.
Python Programs on if else statement		
2	A	1. WAP to check whether the given number is positive or negative.
	A	2. WAP to check whether the given number is odd or even
	A	3. WAP to find out largest number from given two numbers using simple if and ternary operator.
	A	4. WAP to find out largest number from given three numbers.
	A	5. WAP to check whether the given year is leap year or not.
	A	6. WAP in python to display the name of the day according to the number given by the user.
	A	7. WAP to implement simple calculator which performs (addition, subtraction, multiplication and division) of two numbers based on user input.
	B	8. Read marks of five subjects. Calculate percentage and print class accordingly. Fail below 35, Pass Class between 36 to 45, Second Class between 46 to 60, First Class between 61 to 70, Distinction if more than 70.
	B	9. Three sides of a triangle are entered through the keyboard, WAP to check whether the triangle is isosceles, equilateral, scalene or right-angled triangle.
	C	10. Find the second largest number among three user input numbers.

	C	11. WAP to calculate electricity bill based on following criteria. Which takes the unit from the user. i. First 1 to 50 units – Rs. 2.60/unit ii. Next 50 to 100 units – Rs. 3.25/unit iii. Next 100 to 200 units – Rs. 5.26/unit iv. above 200 units – Rs. 8.45/unit
Python Programs on for and while loops		
3	A	1. WAP to print 1 to 10
	A	2. WAP to print 1 to n
	A	3. WAP to print odd numbers between 1 to n
	A	4. WAP to print numbers between two given numbers which is divisible by 2 but not divisible by 3
	A	5. WAP to print sum of 1 to n numbers
	A	6. WAP to print sum of series 1 + 4 + 9 + 16 + 25 + 36 + ...n
	A	7. WAP to print multiplication table of given number.
	A	8. WAP to print sum of series 1 – 2 + 3 – 4 + 5 – 6 + 7 ... n
	A	9. WAP to find factorial of the given number
	B	10. WAP to find whether the given number is prime or not.
	B	11. WAP to find factors of a given number.
	B	12. WAP to print sum of digits of given number
	C	13. WAP to find GCD of the given two numbers.
	C	14. WAP to check whether the given number is palindrome or not
Python Programs on String manipulation		
4	A	1. WAP to check given string is palindrome or not.
	A	2. WAP to reverse the words in given string.
	A	3. WAP to remove i^{th} character from given string.
	A	4. WAP to find length of String without using len() function.
	B	5. WAP to print even length word in string.
	B	6. WAP to count numbers of vowels in given string.
	C	7. WAP to capitalize the first and last character of each word in a string.
	C	8. WAP to convert given array to string.
Python Programs on List		
5	A	1. WAP to find sum of all the elements in a list.
	A	2. WAP to find the largest element in a list.
	A	3. WAP to find the length of a list.
	A	4. WAP to interchange first and last elements in a list.
	A	5. WAP to split the list into two parts and append the first part to the end.
	A	6. WAP to interchange the elements on two positions entered by a user.

	A	7. WAP to reverse a list given by user.
	A	8. WAP to print even numbers in a list.
	B	9. WAP to count unique items in a list.
	B	10. WAP to copy a list.
	B	11. WAP to print all odd numbers in a given range.
	C	12. WAP to count occurrences of an element in a list.
	C	13. WAP to find second largest number in a list.
	C	14. WAP to extract elements with frequency greater than K.
Python Programs on Tuple		
6	A	1. WAP to find sum of tuple elements.
	A	2. WAP to find Maximum and Minimum K elements in a given tuple.
	A	3. WAP to find tuples which have all elements divisible by K from a list of tuples.
	A	4. WAP to create a list of tuples from given list having number and its cube in each tuple.
	A	5. WAP to find tuples with all positive elements from the given list of tuples.
	B	6. WAP to add tuple to list and vice – versa.
	B	7. WAP to remove tuples of length K.
	B	8. WAP to remove duplicates from tuple.
	C	9. WAP to multiply adjacent elements of a tuple and print that resultant tuple.
	C	10. WAP to test if the given tuple is distinct or not.
Python Programs on Set & Dictionary		
7	A	1. WAP to iterate over a set.
	A	2. WAP to convert set into list, string and tuple.
	A	3. WAP to find Maximum and Minimum from a set.
	A	4. WAP to perform union of two sets.
	A	5. WAP to check if two lists have at-least one element common.
	A	6. WAP to sort dictionary by key or value.
	A	7. WAP to find the sum of all items in a dictionary given by user.
	B	8. WAP to check if a given string is binary string or not.
	B	9. WAP to find common elements in three lists using set.
	B	10. WAP to merge two dictionaries given by user.
	C	11. WAP to count number of vowels in given string using set.
	C	12. WAP to handle missing keys in dictionaries.
Python Programs on Functions		
8	A	1. WAP to count simple interest using function.
	A	2. WAP that defines a function to add first n numbers
	A	3. WAP to find maximum number from given two numbers using function.

	A B B C	4. WAP that defines a function which returns 1 if the number is prime otherwise return 0. 5. WAP to generate Fibonacci series of N given number using function name fibbo. (e.g. 0 1 1 2 3 5 8...) 6. WAP to find the factorial of a given number using recursion. 7. WAP to implement simple calculator using lamda function.
Python Programs on File Handling		
9	A A A B B C	1. WAP to read entire fille named abc.txt 2. WAP to print program itself on console. 3. WAP to read first 5 lines from the file named abc.txt 4. WAP to find the longest word in a file named abc.txt 5. WAP to find the size of the file named abc.txt 6. WAP to implement search function to search specific occurrence of word in a given text file.
Python Programs on Exception Handling		
10	A A A B C	1. WAP to handle divide by zero exception. 2. WAP to handle file not found Exception. 3. WAP to handle type Exception. 4. 5. WAP to demonstrate valueError and indexError with example. 5. WAP to raise your custom Exception.
Python Programs on Modules		
11	A A A B C	1. WAP to Pick a random character from a given String. 2. WAP to Pick a random element from a given list. 3. WAP to demonstrate the use of the math module. 4. WAP to demonstrate the use of date time module. 5. WAP to create Calculator module which defines functions like add, sub, mul and div. Create another file that uses the Calculator module.
Python Programs on Graph		
12	A A A A B B C	1. WAP to demonstrate the use of Pie chart. 2. WAP to plot List random of X, Y Coordinates in Matplotlib. 3. WAP to demonstrate the use of Bar chart. 4. WAP to demonstrate the use of Histogram. 5. WAP to display the value of each bar in a bar chart using Matplotlib. 6. WAP create a Scatter Plot with several colours in Matplotlib. 7. WAP to display an Image in Grayscale in Matplotlib.

Python Programs on Object-Oriented Programming		
13	A	1. WAP to create a class by name Students, and initialize attributes like name, age, and grade while creating an object.
	A	2. WAP to create a class named Bank_Account with Account_No, User_Name, Email, Account_Type and Account_Balance data members. Also create a method GetAccountDetails() and DisplayAccountDetails(). Create main method to demonstrate the Bank_Account class.
	A	3. WAP to create Circle class with area and perimeter function to find area and perimeter of circle.
	B	4. WAP to define Time class with hour and minute as data member. Also define addition method to add two-time objects.
	C	5. WAP to demonstrate inheritance in Python.